ISAC (3)

part 3: the software development environment

By Prof. Dr. B. vom Berg, P. Groppe and M. Müller-Aulmann

In this third instalment in the series on the intelligent sensor/actuator controller (or ISAC) we look at the software development environment for the Analog Devices MicroConverter. Following on from the description of the device in part 1 and the construction of a prototyping system in part 2 we are now in a position to write our first program for the ISAC cube.

The Analog Devices MicroConverter can, in principle, be programmed using any of the familiar programming languages, including Assembler51, Basic52, Pascal51, C51 and so on. The result of the translation carried out by the assembler or compiler is simply an Intel Hex file, which must be downloaded, either serially or in parallel, into the MicroConverter's Flash memory.

However, Analog Devices offers a complete software development environment optimised for use with the MicroConverter family, freely downloadable from the Internet: the 'MicroConverter QuickStart Development System V3.0' [1]. This allows programs to be developed in assembler or in C.

Also, a complete 'QuickStart Development Kit' can be ordered from Analog Devices: this contains, in addition to the development software, an ADuC812 evaluation board (in Eurocard format) with mains power supply, data transfer cable and two ADuC812 chips. If you plan to use the hardware described in the last instalment of this series (ISAC cube plus prototyping board), then alongside the software development environment you can find much other information (data sheets, application notes and so on), all freely downloadable from the Internet. **Table 1** and **Table 2** give an overview of the available software tools.

When developing programs for the MicroConverter there are two fundamentally different ways to proceed:

– Develop and test programs using the low-level Assembler51 programming language.

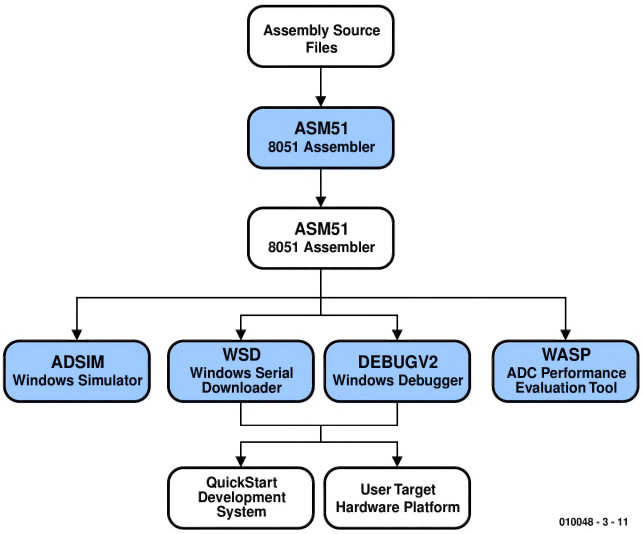

Figure 1. Developing MicroConverter programs in assembler.

– Develop and test programs in a high-level language, i.e. in BASIC52, Pascal51 or C51, of which the last is now regarded as the professionals' choice in industry.

Assembler51

The Assembler51 route is illustrated in **Figure 1**. Using an ordinary text editor the assembler source file is prepared in 8051 assembler code. The Intel Hex file generated with the aid of the 8051 assembler can now be loaded into a simulator (adsim812) running under Windows. This allows the complete behaviour of the MicroConverter to be simulated on the PC at assembler level, including all the on-chip peripherals (A/D, D/A converters and so on), without the need to connect the actual target hardware. Powerful debugging facilities, such as setting breakpoints, single-step execution,

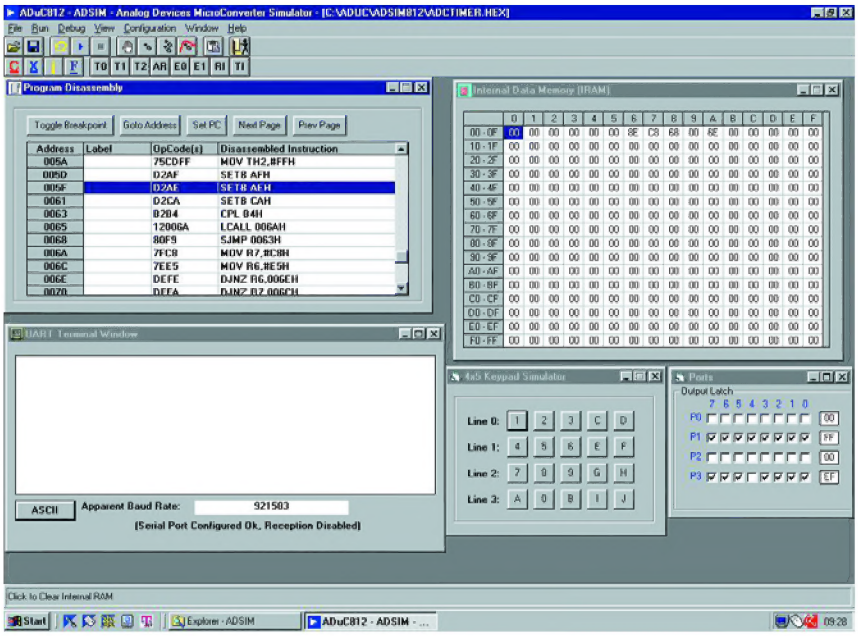

Figure 2. The software-based ADuC812 simulator.

register modification, preloading of data memory and so on are all included in the set of commands. This tool is a purely software-based simulator and debugger, with the whole of the ADuC812 laid out on the PC's screen (**Figure 2**).

If, while simulating or debugging, it is necessary to include the actual target hardware, for example to check the exact behaviour of externally-connected peripheral units (sensors, an LC display, keyboard etc.), then the ADuC Debugger can be used. This connects directly to the development PC using the ADuC812's serial interface. The debug results are taken directly from the actual hardware, and the debugging commands are sent directly to, and carried out by, the actual hardware. Again, all the various test facilities are of course available here: single-step execution, breakpoints, register loading and so on. This tool is thus a hardware-based debugger (**Figure 3**).

In very many cases using the simulator and the debugger will uncover all bugs and problems in a program.

In summary, Assembler51, the software-based simulator, and the hardware-based debugger offer the user an ideal introduction to assembler programming for the MicroConverter family from both a hardware and software point of view.

C51

To develop MicroConverter programs in the C51 high-level language Analog Devices have made available a 'trial' version of the µVision integrated development environment (IDE) produced by Keil, limited to a maximum program size of 2 kbyte. Keil is one of the world's leading producers of IDEs for all derivatives of the 8051 microcontroller family [3]. The µVision user has the great benefit of gaining familiarity with a very powerful professional development tool, with, however, the small concomitant disadvantage that gaining this familiarity takes a little effort (**Figure 4**).

To make the first steps easier, you can download application note uC002 'Writing an ADuC812 Application in C' from the MicroConverter homepage, and application note apnt_151 'Using the Keil Monitor51 /µVision 2 Debugger with the Analog Devices ADuC812 Evaluation Board' from the Keil Homepage (www.keil.com). There

Table 1

The software development environment for the MicroConverter

Programming in a low-level language: Assembler51
MetaLink DOS-based 8051 cross assembler: asm51 (V1.2h).
Windows-based simulator for the ADuC812: adsim812.
Windows-based debugger for the ADuC812: ADuC (V2).

Programming in a high-level language: C51
Keil µVision Integrated Development Environment (IDE) (limited to 2 kbyte maximum program size).

Program downloader
Windows-based serial downloader for programming the on-chip Flash EEPROM program memory from an Intel Hex file and for programming the Flash EEPROM data memory on the device.

Analysis Tools
Windows-based tool for analysis of data values obtained from the A/D converter (including FFT analysis).

Documentation
Getting Started — Introduction
User guides for the MicroConverter
Tool tutorials
Application notes / example code
Data sheets
Broad support via third-party tools

Table 2

Additional C tools for the MicroConverter

Commercial tools (trial versions):
Raisonance Rkit-51 (www.raisonance.com)
IAR Embedded Workbench™ ICC8051 (www.iar.com)

Free tools:
Reads51 (www.rigelcorp.com)
Small Device C Compiler – SDCC (sdcc.sourceforge.net) for Windows and Linux

you will find a very clear step-by-step explanation of how to organise and compile your first project in C51 under µVision, arriving at a downloadable Intel Hex file as the final result. This IDE also allows convenient simulation of your C program. Another reason for downloading and working through these application notes and installing µVision as described is that we will be using C51 exclusively in the following experiments in the world of ADuC812 software.

As already indicated at the start, in addition to the tools described above any other 8051-family programming tools can be used with the MicroConverter. A further selection of tools for programming in the C language can be found in **Table 2**.

Serial Downloader

Now that you have created the final Intel Hex file (containing machine code for the ADuC812), using either Assembler51 or C51 (or some other 8051 language), you must transfer this program code file to the ADuC812.

This can be done with the ADuC debugger, or alternatively you can use the Windows-based Serial Downloader from Analog Devices, designed for the full range of MicroConverter products (**Figure 5**).

This very important process of downloading the program is carried out in three key steps (**Figure 6**).

Step 1: The ADuC812 must first be switched into Download/Debug mode (Figure 6a). To do this, the $\overline{\text{PSEN}}$ pin of the ADuC812 is taken to ground via a 1 kW resistor (resulting in a low level on $\overline{\text{PSEN}}$) and then the reset button is pressed. On the ISAC cube, hold down T2 ('Boot') and then press T1 ('Reset'). Then release T1 and then T2.

Explanation for 8051 experts:

a. *On a 'normal' 8051 the PSEN pin is always an output (read enable for program memory). On the MicroConverter, however, PSEN becomes an input on reset (i.e., as the reset button is released) and the level on the pin is read. If PSEN is at a low logic level, the ADuC812 switches immediately into Download/Debug mode, which means that a program to be downloaded can be received over its serial port and the on-chip Flash program memory can be programmed.*

b. *Each MicroConverter contains a 'secret' program memory in which the manufacturer has provided routines for serial program download, for programming of the Flash EEPROM, and to implement the debugging functions as described above.*

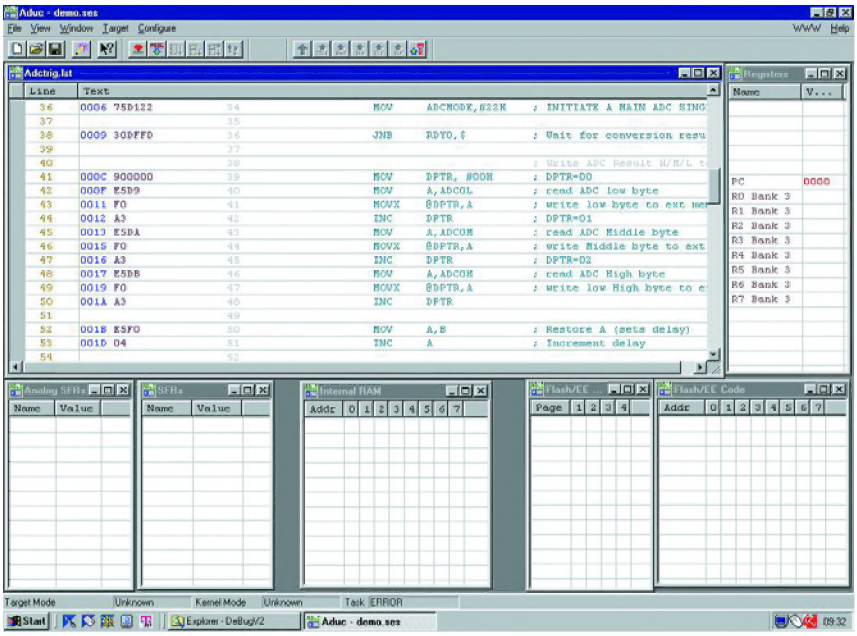

Figure 3. The hardware-based ADuC812 debugger.

On normal initialisation (when power is applied or the reset button is pressed with PSEN not being held low), this special memory area is disabled and invisible: the ADuC812 ignores its contents and runs the application program from Flash EEPROM (normal operation). If, however, the ADuC812 detects a low level on PSEN when the reset button is pressed or released, it ignores the application program in the Flash EEPROM and switches instead to the secret program memory and runs the program stored there — the Download or Debug routine — thereby entering Download/Debug mode.

Step 2. Now you can start the Serial Downloader (**Figure 5**). This program automatically looks for a MicroConverter connected (in the first instance) to COM1, and establishes a connection with it. If the ADuC812

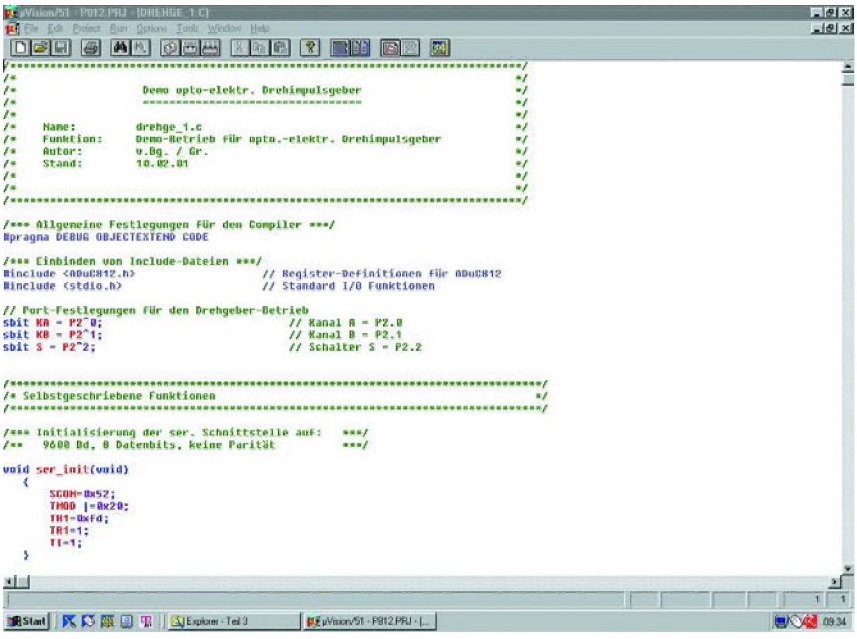

Figure 4. The µVision IDE.

is connected to a different COM interface, the appropriate interface can be chosen using the 'Configuration' menu. Here also you can set other parameters for the download. Initially, leave the settings at their default values. Next click on 'Download' and select, in the usual Windows way, the Intel Hex file to be downloaded from the appropriate directory. Downloading should then proceed automatically and without problems.

Step 3. After the download procedure is complete the Serial Downloader can be dismissed to the task bar. Press the reset button (T1) on the cube, and start your program running on the ADuC812 (**Figure 6b**). Recall that, since you were not holding down T2, PSEN will not be held low and the ADuC812 does not execute from its hidden internal program memory.

Each fresh download proceeds always from step 1 again: hold down T2, press T1, release T1, release T1, start the Serial Downloader, and so on.

Best before date

The Flash EEPROM on the ADuC812 is a non-volatile memory, which means that it retains its contents even in the absence of a power supply — but not forever. Also, there is a limit to how many times fresh programs can be downloaded.

Analog Devices provides the following information on this point in the ADuC812 data sheet:

- Data are retained in the Flash memory (with or without power supply) for at least 10 years; after that there may be 'drop-outs' which manifest themselves as errors in the stored data.
- The maximum number of reprogramming cycles (erase and rewrite of memory locations before errors occur in the memory) is given as 10,000.

It is worth noting that in each ADuC812 there are two separate regions of Flash memory available.

For the program memory the 10,000 reprogramming cycles are certainly enough: if you rewrite your program 20 times a day, you can carry on for a full 500 days! Things are different, however, with the 640-byte Flash EEPROM data memory, where any desired data can be written as often as required. In some cases it is possible to reach or even overstep the critical limit of 10,000 erase/write cycles very quickly, especially when, for example, you store frequently changing measured values in these memory locations. In such cases it is better to opt for external expansion data memory in the form of a RAM (how this can be done will of course be explained later). But even here Analog Devices is developing its MicroConverter product line further: later derivatives (see Table 2 in the first instalment of this series) have a data retention of up to 100 years(!) and are specified for at least 100,000 erase/write cycles.

Emulators

Finally we should note that there are emulators available for the MicroConverter. MetaLink (*www.metaice.com*) produces a classical emulator for the ADuC812: that is, one which replaces the MicroConverter in its socket in the target system.

Accutron (*www.accutron.com*) produces the ACE 'single pin emulator', which supports all derivatives in the MicroConverter family and which includes C source-level debugging. The emulator is connected to the target system via the EA pin. Like the PSEN pin, this pin has many uses, which are described in the data sheet and in the MicroConverter 'User's Manual'. In regard to MicroConverter derivatives with 16- and 24-bit analogue-to-digital converters in particular, this type of connection with the target system is of course very simple and has many advantages. This puts a powerful development tool at the disposal of the professional user.

Once you have learnt how to use the full range of tools to turn your ISAC cube programming ideas into reality, we can proceed in the next instalment to a few suggestions for interesting applications written in C51. Meanwhile, Analog Devices is developing an enhanced version of the development tools for the MicroConverter, and so it is worth keeping an eye on the MicroConverter homepage (*www.analog.com/microconverter*), to obtain the most up-to-date information.

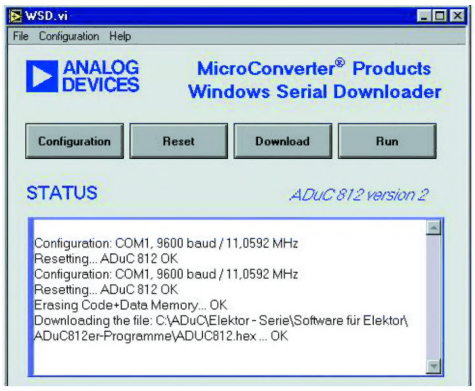

Figure 5. The Serial Downloader.

(010048-3)

Literature and websites:

[1] MicroConverter Internet homepage: *www.analog.com/microconverter*
[2] Analog Devices on the Internet: *www.analog.com*
[3] Keil µVision IDE: *www.keil.com*

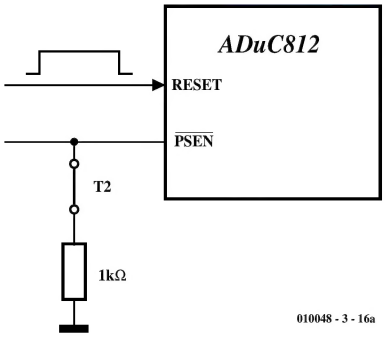

Figure 6a. Activation of debug/program download mode.

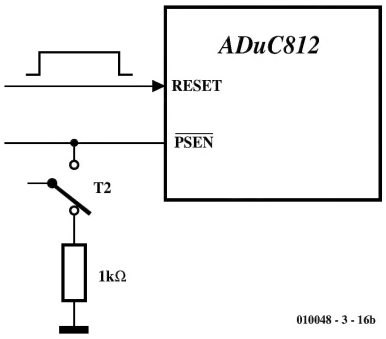

Figure 6b. Normal operating mode.